**RESTRICTED ACCESS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DISTRIBUTION :** | **Firm** | **To** | **Ref** | **Copies** | **1st page** | **e-mail** |
|  | FCE | Xavier Christmann, Claude Redon, Adel Bassaid | 8 |  |  |  |
|  | FCE | Hamza Zetti, Lory Médas, Sabrine Bouazizi, Wail Amri, Kérima Adjadi, Mickael Pastor, Aki Saito | 10 |  |  |  |
|  | RBE | S. Papadineti, D. Andris, M. Ianos, M. Obada, A. Negrea, S. Dominte | 6 |  |  |  |
|  |  |  |  |  |  |  |
|  | FCE | Secretary ship | 1 |  |  |  |

**SW Architecture Design & Interface Description :**

**BFE sw UNIT**

OBJECT: This document is the description of the design & interfaces for *BFE* SW unit.

SUMMARY: This document provides a high-level view of the *BFE* SW unit. The inputs of this document are provided by the software requirement. It is linked to the DAIMLER\_MMA\_SWarchitectureDesignInterfaceDescription document.

CONCLUSION: Applicable from R01.0 SW release

**THIS DOCUMENT CONTAINS HIDDEN TEXT**

EVOLUTION OF THE DOCUMENT

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Author** | **Motive and nature of the modifications** |
| 1 | 31/08/2016 | C. Redon | First release (extract from the full PP4G architecture document) |
| Start extended description based on mainstream document | | | |
| 1.1.1.2 | 10/07/2019 | A. Vaché | Update traceability to match PP4G extended platform requirements IDs |
| 1.1.1.3 | 29/07/2019 | A. Vaché | Remove traceability to no more existing requirements |
| 1.1.1.4 | 08/08/2019 | A. Vaché | Solve some traceability issues highlighted by reqtify |
| 1.1.1.5 | 24/01/2020 | A. Vaché | Rearrange some chapters related to Dataflow |
| 1.1.1.6 | 02/03/2020 | A. Diankouika | Update distribution list |
| 1.1.1.7 | 24/04/2020 | A. Diankouika | Add traceability for ARCH\_SW\_BFE\_00300 |
| 1.1.1.8 | 24/04/2020 | A. Diankouika | Add ARCH\_SW\_BFE\_0392 |
| Start DAI mmadescription based on extended document | | | |
| 1.1.1.5.1.1 | 06/01/2022 | A. Negrea | First revision |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

This document contains **34** pages.

**CONTENTS**

1. Documentation 6

1.1. Upper Level Relevant Documents 6

1.2. Design Specification Documents 7

1.3. Freescale Documents 8

1.4. Tier2 Documents 9

1.5. HW Datasheet 9

1.6. Other Documents 9

1.7. Glossary And Definition 10

2. Description 11

3. Technical functions 11

3.1. Main functions 12

3.2. To Start/Stop/Abort a belt function (aka cycle) 12

3.2.1. To start a belt function 13

3.2.2. To stop a belt function 14

3.2.3. To abort a belt function 15

3.3. To prepare the belt function execution 16

3.4. To execute the belt function 17

3.4.1. Principle 17

3.4.2. To execute the “standard” step 19

3.4.3. To execute the “high power” (booster) step 20

3.5. To execute the emergency motor braking sequence (eCPL function) 22

3.5.1. Principle 22

3.5.2. To disable the belt function activation 22

3.5.3. To brake the motor rotation 23

3.6. To update the power supply under critical section 24

4. Runnables 26

4.1. BFE\_runScheduleStep 26

4.1.1. Definition 26

4.1.2. Data flow 26

4.1.3. Called functions 28

4.2. BFE\_runExecuteStandardSteps 29

4.2.1. Definition 29

4.2.2. Data flow 29

4.2.3. Called functions 30

4.3. BFE\_runExecuteHighPowerStep 30

4.3.1. Definition 30

4.3.2. Data flow 31

4.3.3. Called functions 31

4.3.4. Miscelaneous 32

4.4. BFE\_runExecuteEmergencyBrakingSequence 32

4.4.1. Definition 32

4.4.2. Data flow 32

4.4.3. Called functions 32

4.5. BFE\_runResetMotorStageStatus 33

4.5.1. Definition 33

4.5.2. Data flow 33

4.5.3. Called functions 33

5. MCU resources 33

**TABLE OF FIGURES**

[**Figure 1: Bfe - Static description** 11](#_Toc62556908)

[**Figure 2: Bfe – Overview of main functions** 12](#_Toc62556909)

[**Figure 3: Bfe - Data flow description to start a belt function** 13](#_Toc62556910)

[**Figure 4: Bfe - Data flow description to stop a belt function** 14](#_Toc62556911)

[**Figure 5: Bfe - Data flow description to abort a belt function** 15](#_Toc62556912)

[**Figure 6: Bfe - Data flow description to prepare the belt function execution** 16](#_Toc62556913)

[**Figure 7: Bfe - Dynamic description to execute a cycle without booster function** 18](#_Toc62556914)

[**Figure 8: Bfe - Data flow description to execute a non-high power step** 19](#_Toc62556915)

[**Figure 9: Bfe - Data flow description to execute the high power step** 20](#_Toc62556916)

[**Figure 10: Bfe - Dynamic description to execute a cycle with booster function** 21](#_Toc62556917)

[**Figure 11: Bfe – To disable the belt functions** 22](#_Toc62556918)

[**Figure 12: Bfe - To execute the emergency motor braking sequence** 23](#_Toc62556919)

[**Figure 13: Bfe - To update the motor power order under critical section – Case #1** 24](#_Toc62556920)

[**Figure 14: Bfe - To update the motor power order under critical section – Case #2** 25](#_Toc62556921)

# Documentation

## Upper Level Relevant Documents

This section presents all the documents needed to write the software architecture design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
|  | TF-A: To Manage the power supply | /RevAS/30\_DES\_Requirements/Technical Functions/  DES\_TF\_A\_To\_Manage\_The\_Power\_Supply | RBE/FCE |
|  | TF-B: To Manage the communication | /RevAS/30\_DES\_Requirements/Technical Functions/  DES\_TF\_B\_To\_Manage\_The\_Communication | RBE/FCE |
|  | TF-C: To Secure PP ECU functioning using Pictus MCU | /RevAS/30\_DES\_Requirements/Technical Functions/  DES\_TF\_C\_To\_Secure\_PP\_ECU\_Functioning\_Pictus | RBE/FCE |
|  | TF-D: To Program MCU | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_D\_To\_Program\_MCU | RBE/FCE |
|  | TF-E: To Manage Diagnostic Requests | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_E\_To\_Manage\_Diagnostic\_Requests | RBE/FCE |
|  | TF-F: To Perform Measurements | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_F\_To\_Perform\_Measurements | RBE/FCE |
|  | TF-G: To Drive the Motor | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_G\_To\_Drive\_the\_Motor | RBE/FCE |
|  | TF-H: To Perform Autotests | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_H\_To\_Perform\_Autotests | RBE/FCE |
|  | TF-I: To Manage the Failure | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_I\_To\_Manage\_The\_Failure | RBE/FCE |
|  | TF-J: To Manage NVM - NVP (Non Volatile Parameters) | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_J\_To\_Manage\_NVM | RBE/FCE |
|  | TF-K: To Ensure ECU Protection and Integration | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_K\_To\_Ensure\_ECU\_Protection\_And\_Integration | RBE/FCE |
|  | TF-L: To Ensure ECU Integration in Environment EMC ESD | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_L\_To\_Ensure\_ECU\_Integration\_In\_Environment\_EMC\_ESD | RBE/FCE |
|  | TF-M: To generate time base | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_M\_To\_Generate\_Time\_Base | RBE/FCE |
|  | TF-N: To evaluate belt data | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_N\_To\_Evaluate\_Belt\_Data | RBE/FCE |
|  | TF-O: To schedule the SW | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_O\_To\_Run\_SW | RBE/FCE |
|  | TF-P: To handle network management | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_P\_To Handle\_Network\_Management | RBE/FCE |
|  | TF-Q: To Provide Data For Expertise | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_Q\_To\_Provide\_Data\_For\_Expertise | RBE/FCE |
|  | TF-R: To Decide Belt Function Execution | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_R\_To\_Decide\_Belt\_Function\_Execution | RBE/FCE |
|  | TF-S: To drive the boost | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_S\_To\_Drive\_Boost | RBE/FCE |
|  | TF-X: To generate time base | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_M\_To\_Generate\_Time\_Base | RBE/FCE |

## Design Specification Documents

This section presents all the documents that complete this software architecture design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
|  | EEPROM parameters | SBE\_4G\_NVP\_layout.xls | RBE/FCE |
|  | Design Interface description of AdcIf | N/A | RBE/FCE |
|  | Design Interface Description of Auto Tests Manager | N/A | RBE/FCE |
|  | Design Interface Description of Belt Function Decision | N/A | RBE/FCE |
|  | Design Interface Description of Belt Function Execution | [BFE - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFE%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design Interface Description of Belt Function Selection | [BFS - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFS%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design Interface Description of Belt Movement Monitoring | N/A | RBE/FCE |
|  | Design Interface Description of Belt Parking Algorithm | N/A | RBE/FCE |
|  | Design Interface Description of Belt Slack Reduction | N/A | RBE/FCE |
|  | Design Interface Description of Basic Software Manager | N/A | RBE/FCE |
|  | Design Interface Description of Basic Software Manager Interface | N/A | RBE/FCE |
|  | Design Interface Description of Can Tranceiver Interface | N/A | RBE/FCE |
|  | Design Interface Description of Communication Interaction Layer | [CIL - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\CIL%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Diagnostic Communication Manager Interface | N/A | RBE/FCE |
|  | Design Interface Description of Diagnostic Event Manager Interface | N/A | RBE/FCE |
|  | Design Interface Description of DiagOnCAN services management | [DIA - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\DIA%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Electronic Control Unit Manager | N/A | RBE/FCE |
|  | Design Interface Description of Electronic Control Unit Manager Interface | N/A | RBE/FCE |
|  | Design Interface Description of End of life | N/A | RBE/FCE |
|  | Design Interface Description of Error Handler | N/A | RBE/FCE |
|  | Design Interface Description of Haptic Warning | N/A | RBE/FCE |
|  | Design Interface Description of Memory Integrity Control | N/A | RBE/FCE |
|  | Design Interface Description of Mode Management | [MMG - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\MMG%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Network Management Interface | N/A | RBE/FCE |
|  | Design Interface Description of Non-Volatile Memory Interface | N/A | RBE/FCE |
|  | Design Interface Description of Non-Volatile Parameters | [NVP - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\NVP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Operating System Interface | N/A | RBE/FCE |
|  | Design Interface Description of Power Abstraction Layer | [PAL - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PAL%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Pre-Crash Master | N/A | RBE/FCE |
|  | Design Interface Description of Physical Measures Provider | [PMP - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PMP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Port Interface | N/A | RBE/FCE |
|  | Design Interface Description of Pre Pre-Tensioning | [PRE - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PMP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Production cycle function | N/A | RBE/FCE |
|  | Design Interface Description of Pulse Width Modulation Interface | N/A | RBE/FCE |
|  | Design Interface Description of Reset Cause Management | N/A | RBE/FCE |
|  | Design Interface Description of SBC | N/A | RBE/FCE |
|  | Design Interface Description of System Context Management | N/A | RBE/FCE |
|  | Design Interface Description of Standard Function Recovery (releasing function) | [SFR - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\SFR%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Serial Peripheral Interface Interface | N/A | RBE/FCE |
|  | Design Interface Description of Startup | N/A | RBE/FCE |
|  | Design Interface Description of System Time Management | N/A | RBE/FCE |
|  | Design Interface Description of Vehicle Dynamics algorithm | N/A | RBE/FCE |

## Freescale Documents

This section presents all the documents that complete this software architecture design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
|  | MC9S12ZVC-Family Reference Manual Preliminary  Confidential | MC9S12ZVCRM\_Rev0.06.pdf | Freescale |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Tier2 Documents

This section presents all the documents that complete this software architecture design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## HW Datasheet

This section presents all the documents related to the HW components that complete this software architecture design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
|  | BTN8984TA datasheet | BTN8984TA\_TDS\_051 | Infineon |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Other Documents

This section presents all the documents that also have been needed to write this software architecture design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
|  | Unified Modelling Language | 2.1.1 | OMG |
|  | MCU RFQ | [E2581849](https://plm.autoliv.int/linkto/latest/ProductDescription/E2581849/*) | FCE |
|  |  |  |  |

## Glossary And Definition

This section presents all the definitions and/or abbreviations used in this document.

*List of terms in alphabetical order:*

|  |  |
| --- | --- |
| ***Term*** | ***Meaning*** |
| ADC | Analog Digital Converter |
| AEC | Autoliv Error Code |
| API | Application Programming Interface |
| ASDM | Active Safety Domain Master |
| ASIC | Application Specific Integrated Circuit |
| ASY | Active SafetY |
| BSW | Basic SW modules |
| CAN | Controller Area Network |
| C/S | Chip Select |
| COP | Computer Operating Properly |
| eCPL | Electronic Crash Pole Locking |
| DART | Ditch - Airborne - Rough Terrain |
| DFLASH | Data FLASH |
| ECC | Error Code Correction |
| ECU | Electronic Control Unit |
| EOL | End Of Life |
| EEPROM | Electric Erasable and Programmable Read only Memory |
| HFPP | High Force Pre-Pre-Tensioning belt function |
| HF-PRE | High Force PRE pre-tensioning |
| HR | Hard Releasing |
| I/O | Input/Output |
| IMU | Inartial Measurements Unit |
| ISS | Integrated Safing System |
| LFPP | Low Force Pre-Pre-Tensioning belt function |
| MSA | Motor Start/Stop Automatic |
| MCAL | Micro-Controller Abstraction Layer |
| MCU | Micro-controller Unit |
| NMG | Mode ManaGement |
| NVM | Non Volatile Memory |
| OS | Operating System |
| PCM | Pre-Crash Master |
| PFLASH | Program FLASH |
| PIT | Periodic Interrupt Timer |
| PLL | Phase-locked loop |
| RAM | Random Access Memory |
| RCWM | Rear Collision Warning and Mitigation |
| RML | Left PP ECU |
| RMR | Right PP ECU |
| RMx | Both PP ECU |
| ROM | Read Only Memory |
| RSU | Remote Sensor Unit |
| RTE | Real Time Environment |
| RTOS | Real Time Operating System |
| SFR | Standard Function Recovery |
| SODL | Side Obstacle Detection Left |
| SPI | Serial Peripheral Interface |
| SRS | Supplementary Restraint System |
| TBC | To be confirmed |
| TBD | To be defined |
| TF | Technical Function |
| TFLASH | Test FLASH of the Pictus MCU (“one time programmable” memory) |
| W/D | Watchdog |

# Description

The BFE component is the main component of the SW application. It is responsible for executing the belt functions selected by BFS.

# Technical functions

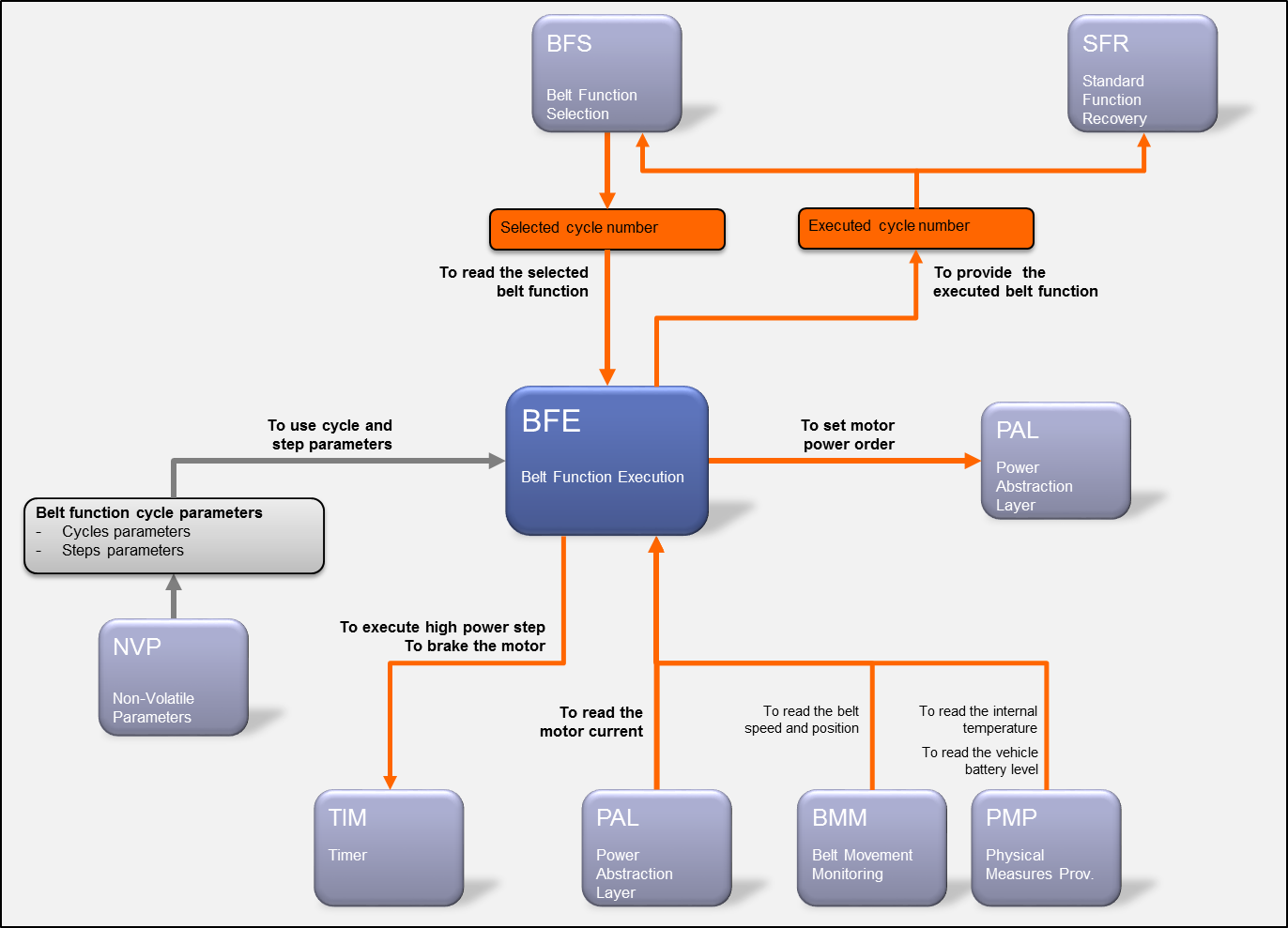
|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_0000 | This component shall implement the TF-G technical function. Refer to [A6] for more details. |  |  |



The following figure shows the static description of the module.

As depicted by this figure, the BFE component is connected to many components which can be grouped in 2 categories:

1. ECU abstraction layer
2. Belt function decision and selection algorithm



**Figure 1: Bfe - Static description**

## Main functions

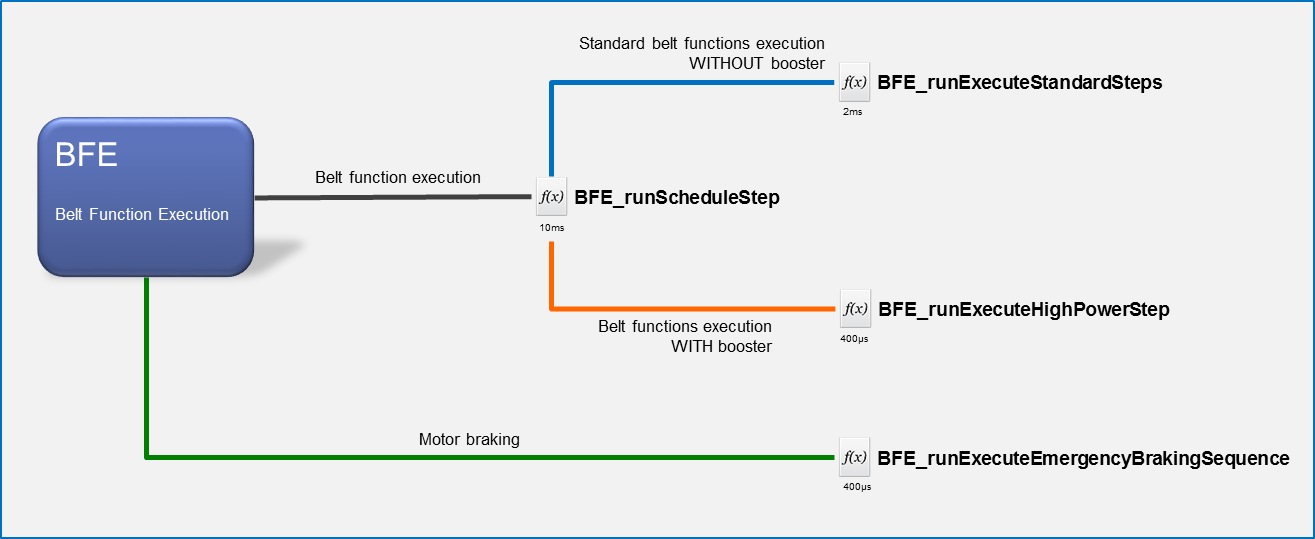
BFE is composed of 4 main periodic functions:

For the belt function execution

* BFE\_runScheduleStep
* BFE\_runExecuteStandardSteps
* BFE\_runExecuteHighPowerStep

For the motor braking sequence execution

* BFE\_runExecuteEmergencyBrakingSequence



**Figure 2: Bfe – Overview of main functions**

## To Start/Stop/Abort a belt function (aka cycle)

Before describing the execution of a selected cycle the present section will make a focus on the 2 main data of the SW:

* The “selected” cycle number
* The “executed” cycle number

These 2 data are the basis to start, stop and abort a belt function execution.

Most of the time, these 2 data will have the same value; BFE will execute what BFS selects.

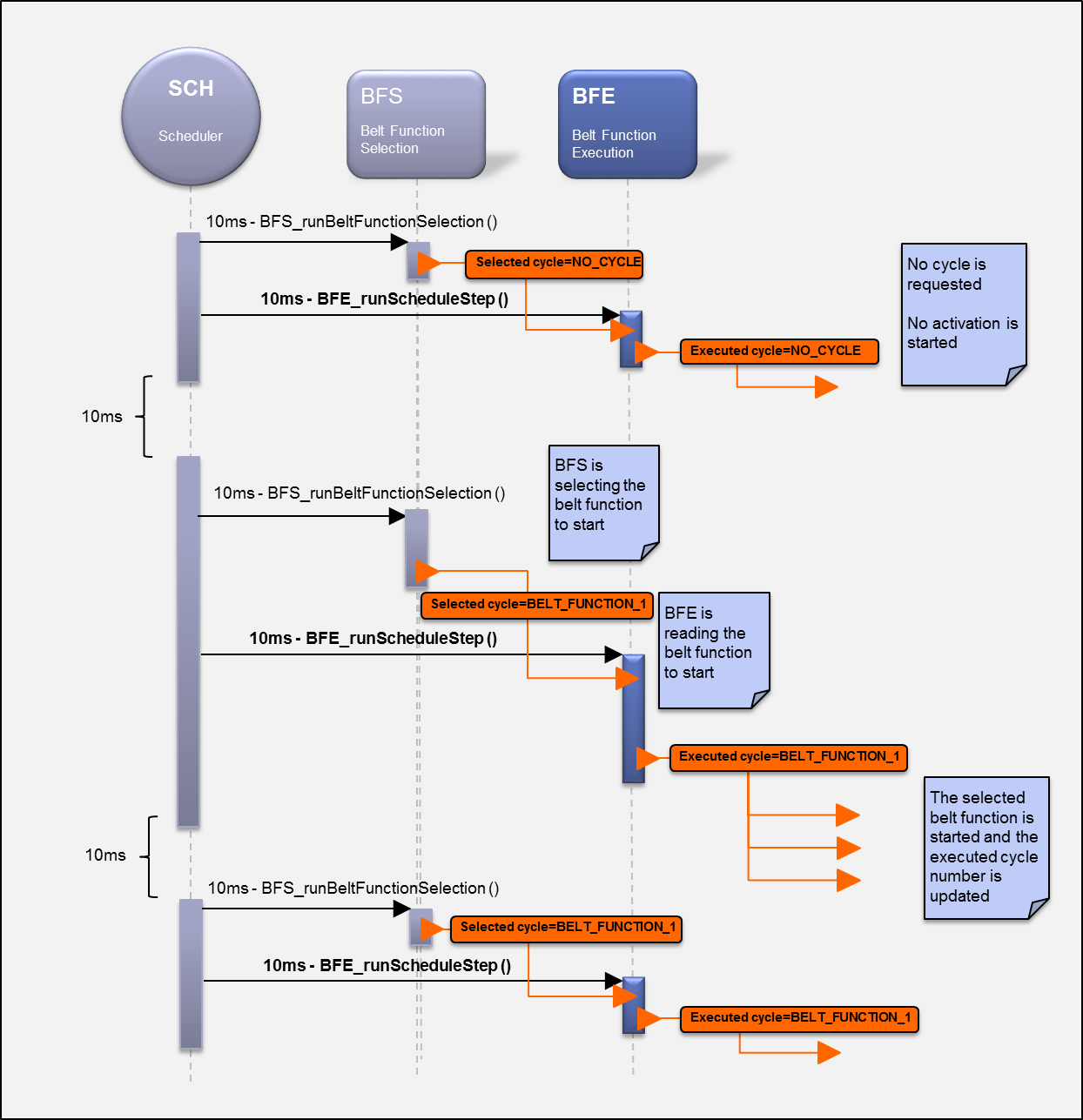
Nevertheless the value of these values can differ when belt function is started, terminated or aborted.

The purpose of the next sub-chapters is to explain the basis regarding the functioning of these 2 data.

### To start a belt function

As depicted by the figure below, the start of a belt function (aka cycle) is based on the “selected cycle” data provided by BFS:

* By default “NO\_CYCLE” is selected by BFS (no belt function is requested).
* When belt functions are requested BFS will select the belt function having the highest priority. The selected belt function will be then provided to BFE.
* BFE will start the execution of the selected cycle.



**Figure 3: Bfe - Data flow description to start a belt function**

### To stop a belt function

The nominal ending of a belt function execution is driven by BFE itself.

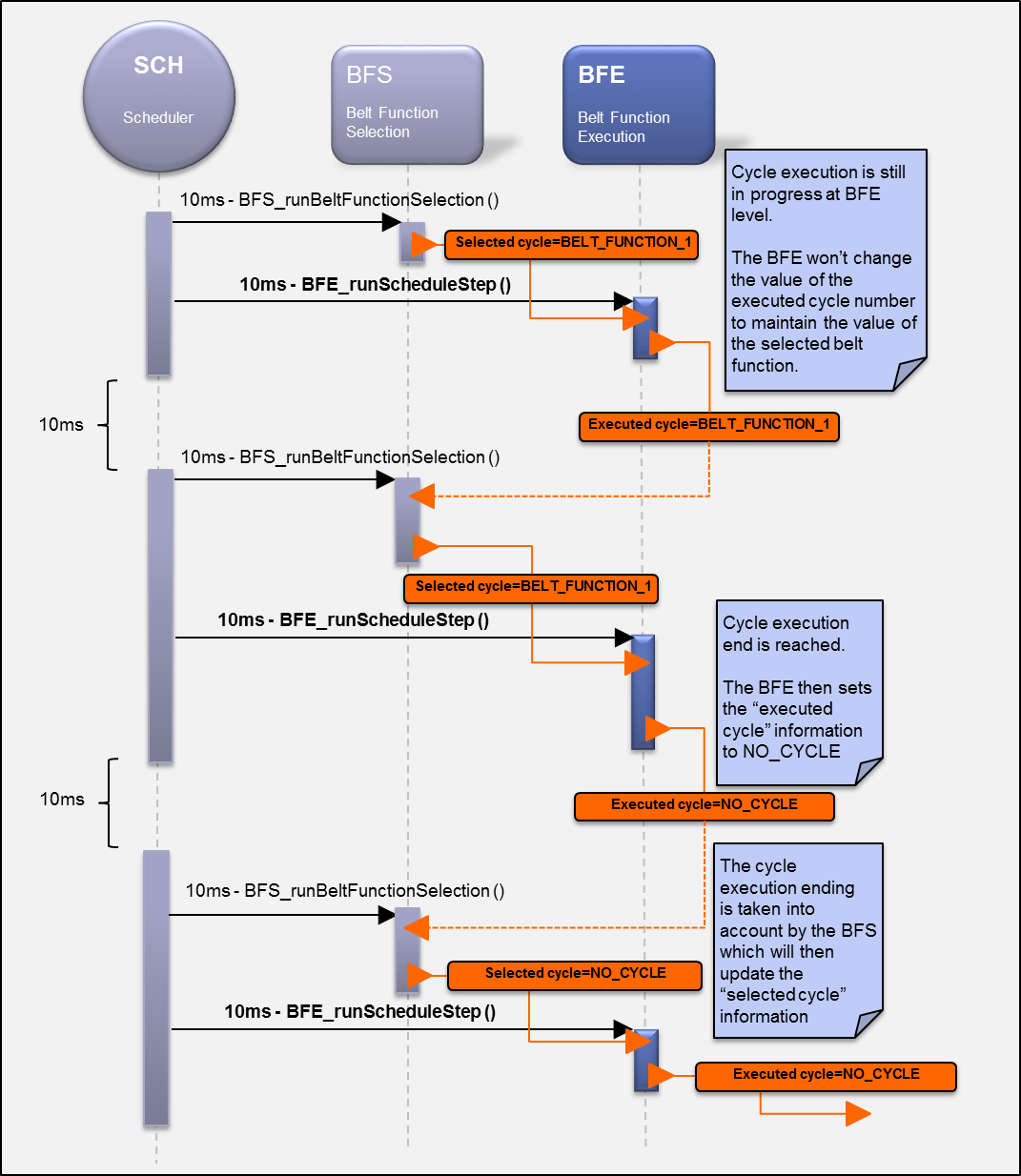
At the end of the belt function execution, BFE will set the “executed cycle” information to NO\_CYCLE.

This change will be taken into account during the next execution of the BFS main function, leading to select NO\_CYCLE

Note:

The value of the executed belt function is maintained by BFE until the end of the belt function execution.

BFS will then continue to select and provide this current executed cycle to BFE except if another belt function, having a higher priority, is requested.



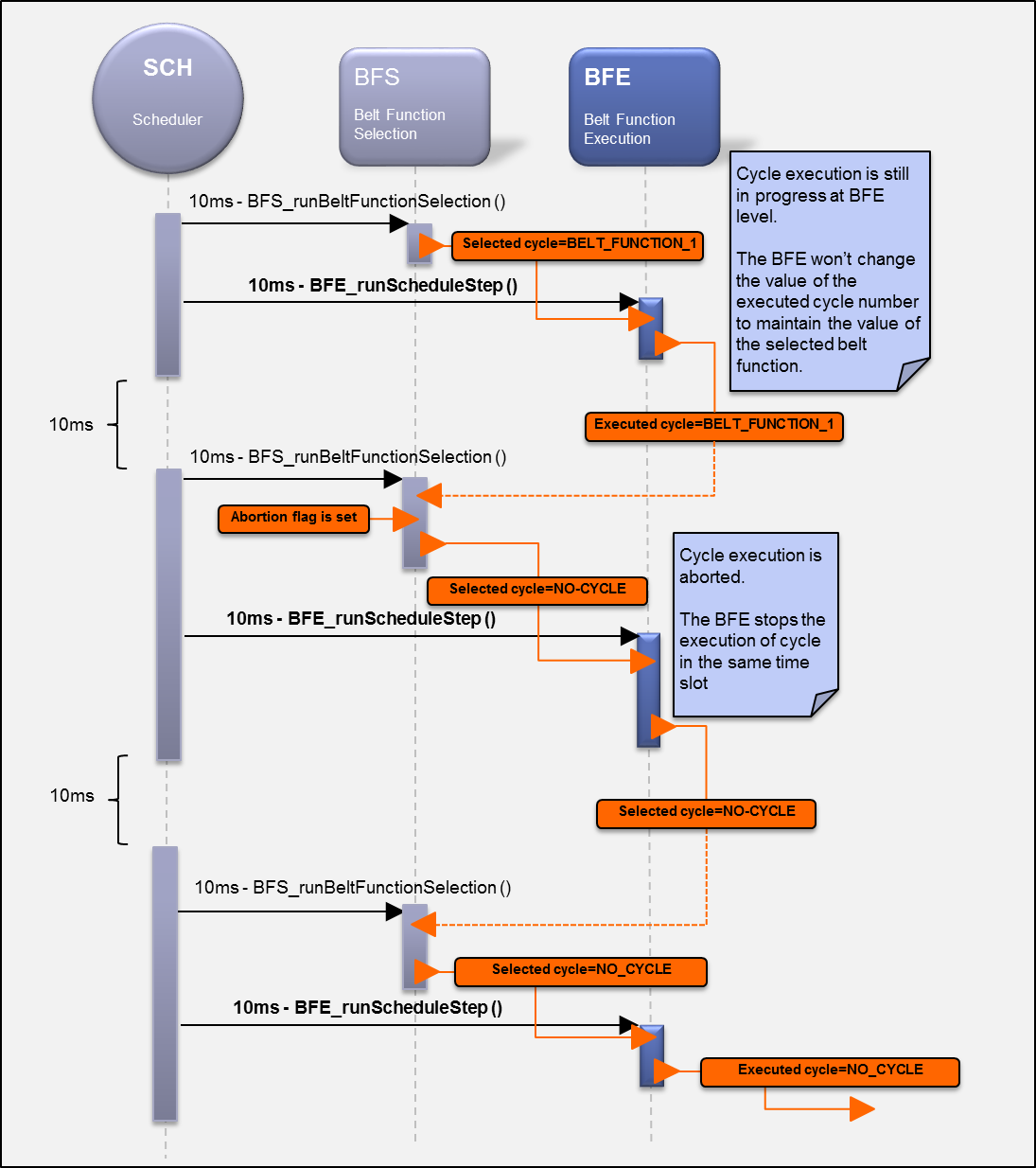
**Figure 4: Bfe - Data flow description to stop a belt function**

### To abort a belt function

On the contrary of the nominal belt function execution ending, the interrupt of a belt function is driven by BFS.

When BFS receives an abortion/interrupt request (from a decision algorithm) for the current executed cycle, BFS will select and provide NO\_CYCLE to BFE. Then BFE will immediately stop the current executed cycle.

Both “selected” and “executed” cycle will revert to their default value (NO\_CYCLE) at the same time (from an execution time-slot point of view)



**Figure 5: Bfe - Data flow description to abort a belt function**

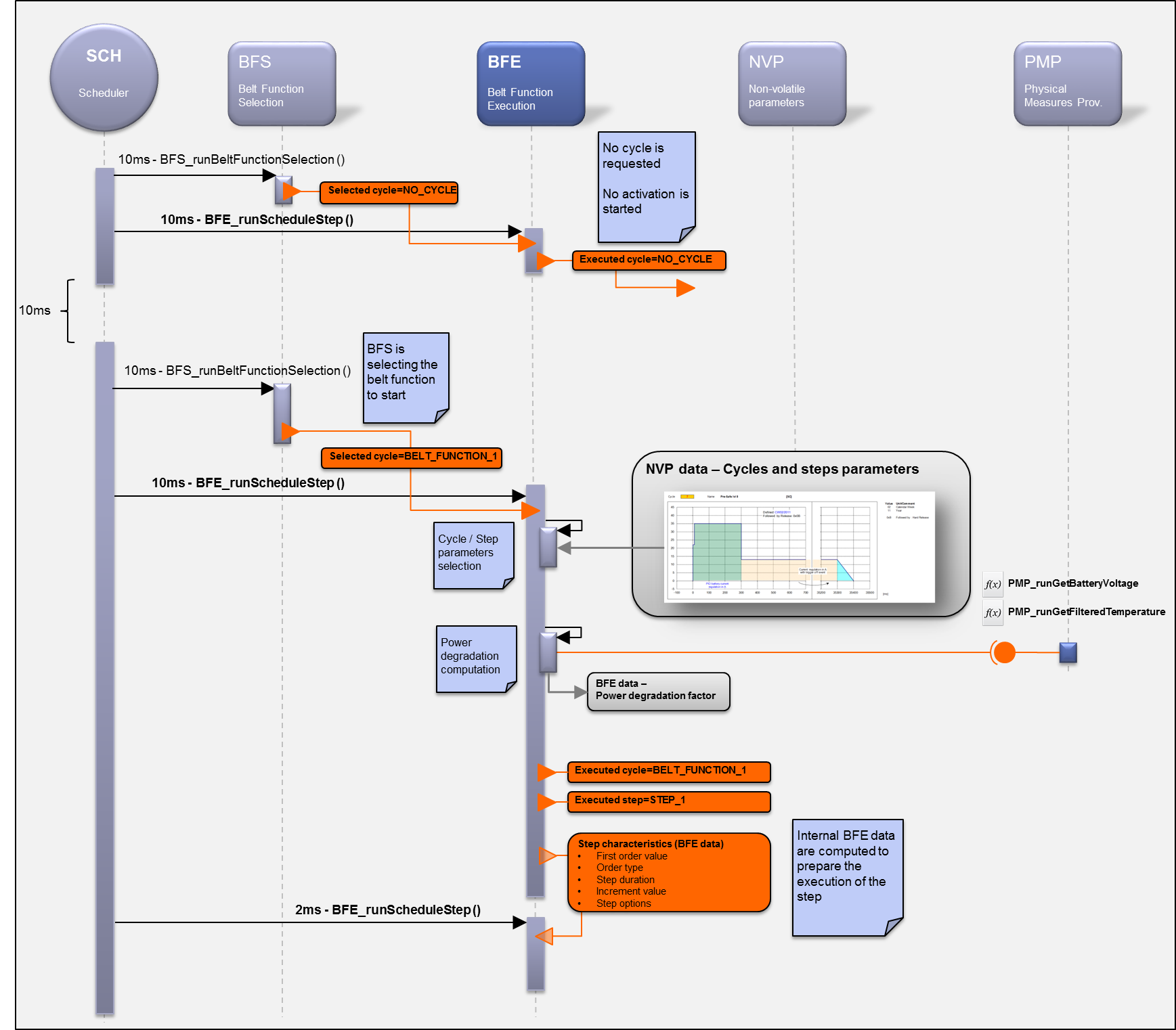
## To prepare the belt function execution

Once the start of the belt function has been decided, BFE will perform the following tasks to prepare the execution:

* To select the cycle parameters corresponding to the belt function to execute
* To take into account the battery voltage for the power degradation function
* To take into account the system temperature (measured at ECU level) for temperature compensation function

All these actions will be performed one time by the BFE\_runScheduleSteps function, at the beginning of belt function execution. Refer to [A6] for more details about the functional aspect.

Once these actions are performed, BFE will start the execution of the belt function.



**Figure 6: Bfe - Data flow description to prepare the belt function execution**

## To execute the belt function

### Principle

The execution of a belt function is based on 3 main functions:

* BFE\_runScheduleStep:
  + To schedule the steps of the selected belt function profile (cycle parameters)
  + Execution period: 10ms
* BFE\_runExecuteStandardSteps
  + To execute the step WITHOUT booster function
  + Execution period: 2ms
* BFE\_runExecuteHighPowerStep
  + To execute the step WITH booster function
  + Execution period: 400µs

The 2 last functions will periodically compute and provide a motor power order to PAL in order to drive the motor in both directions.This will be based on:

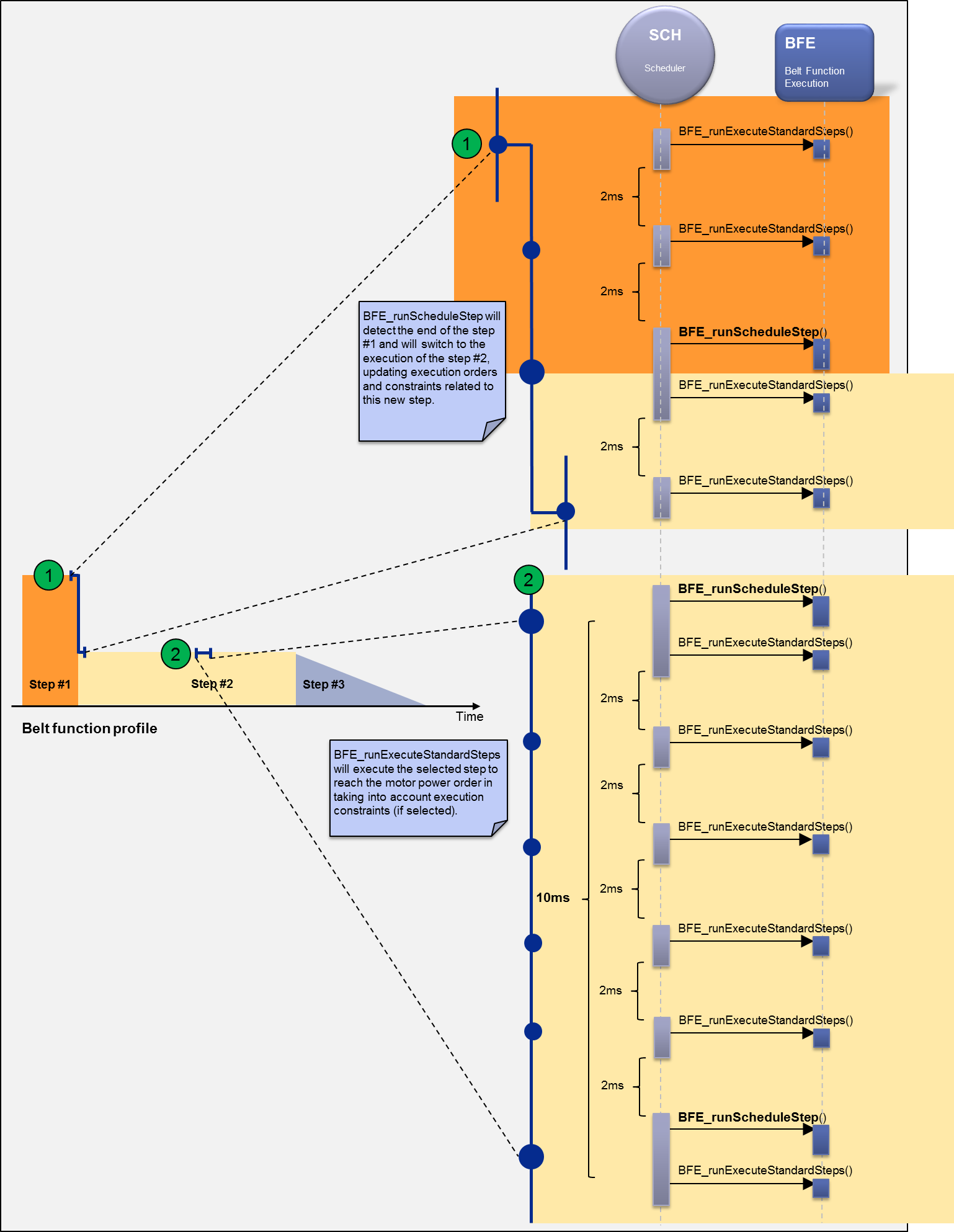
* Cycle parameters
  + The step type
  + The step execution constraints
  + The motor control setpoint
* Input data
  + Physical measurements
  + Algorithm outputs
  + CAN signal

Note:

The computation method of the motor power order depends on the step type. Refer to [A6] for more details.

The figure at the next page illustrates an example of the execution of the belt function when:

* BFE selects the next step
* BFE executes a step WITHOUT the booster function



**Figure 7: Bfe - Dynamic description to execute a cycle without booster function**

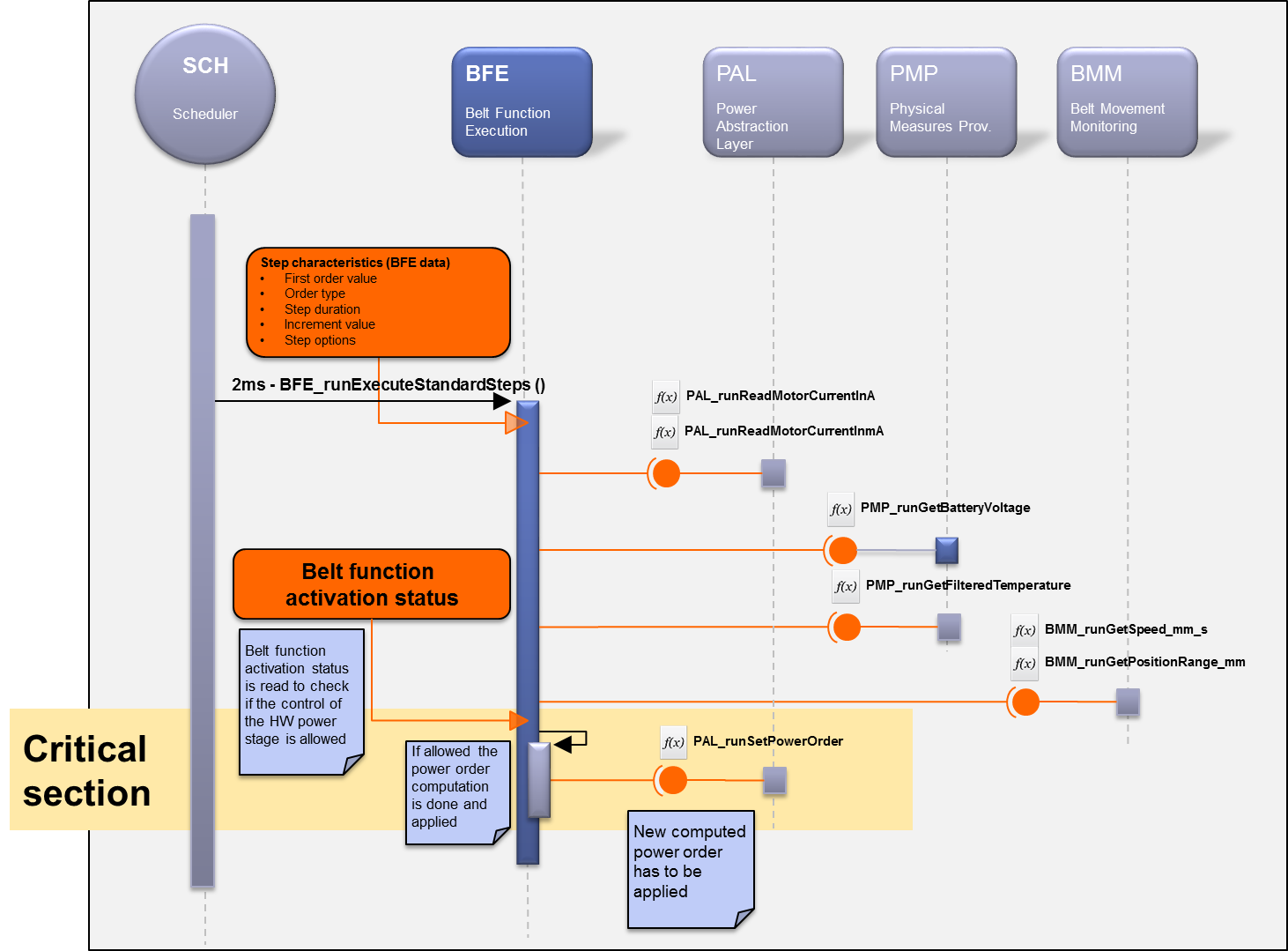
### To execute the “standard” step

This new section makes a focus on the execution of a standard step (booster function NOT used).

As depicted by the figure below, the BFE\_runExecuteStandardSteps function will be called every 2ms by the scheduler. Depending on the step type this function will take into account specific measurements (current, voltage…) to compute and provide the new motor power order to PAL.

Note:

Refer to [[A6]](#_Hlk442359096) for more details about step types.

****

**Figure 8: Bfe - Data flow description to execute a non-high power step**

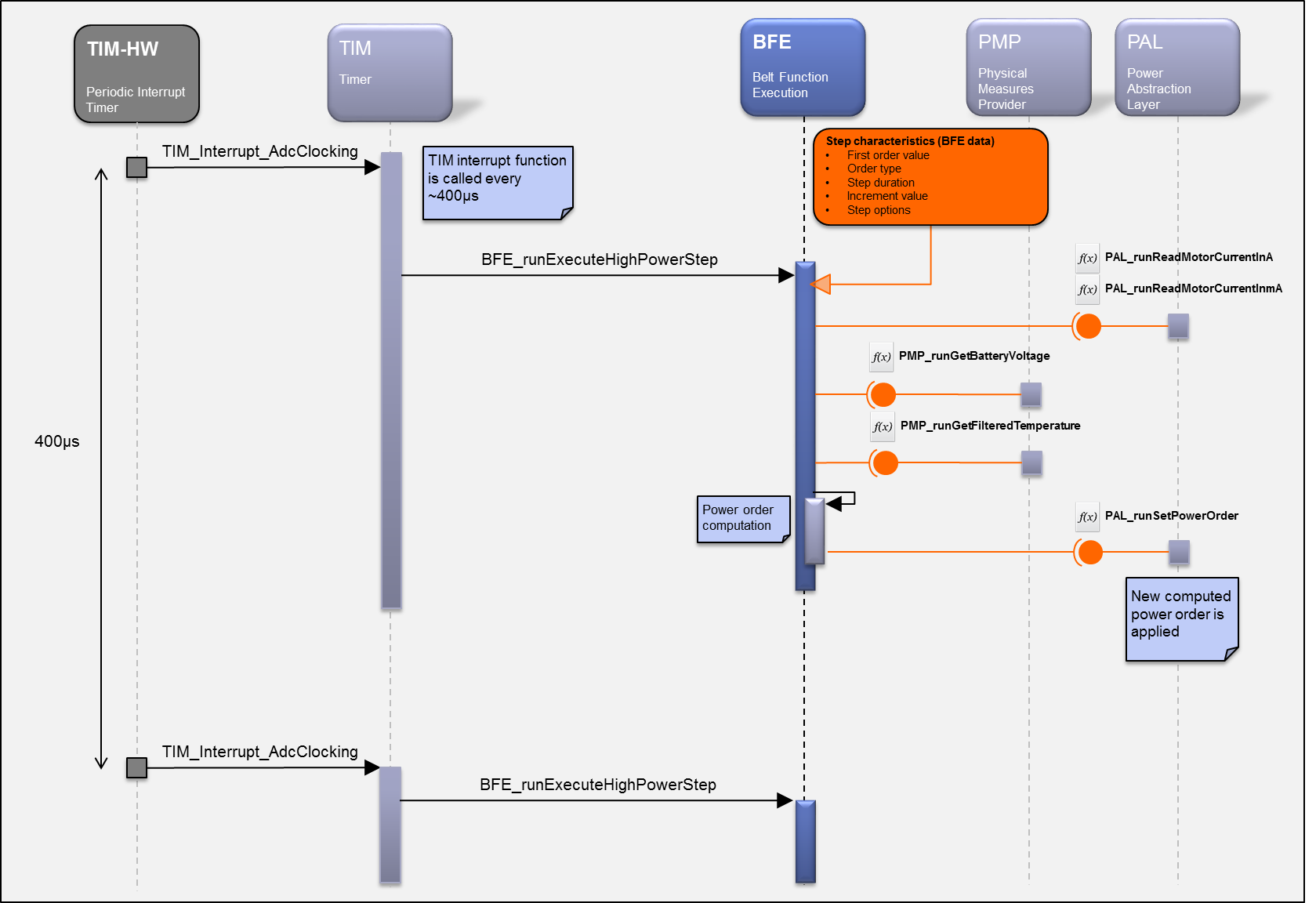
### To execute the “high power” (booster) step

The principle of the “high power” step execution (specific option in the step parameters) is the similar to the “standard” one.

However the high power function has 2 constraints:

1. The current loop control period has to be faster to the standard one
2. The current loop control execution has to be close to the motor current sampling point

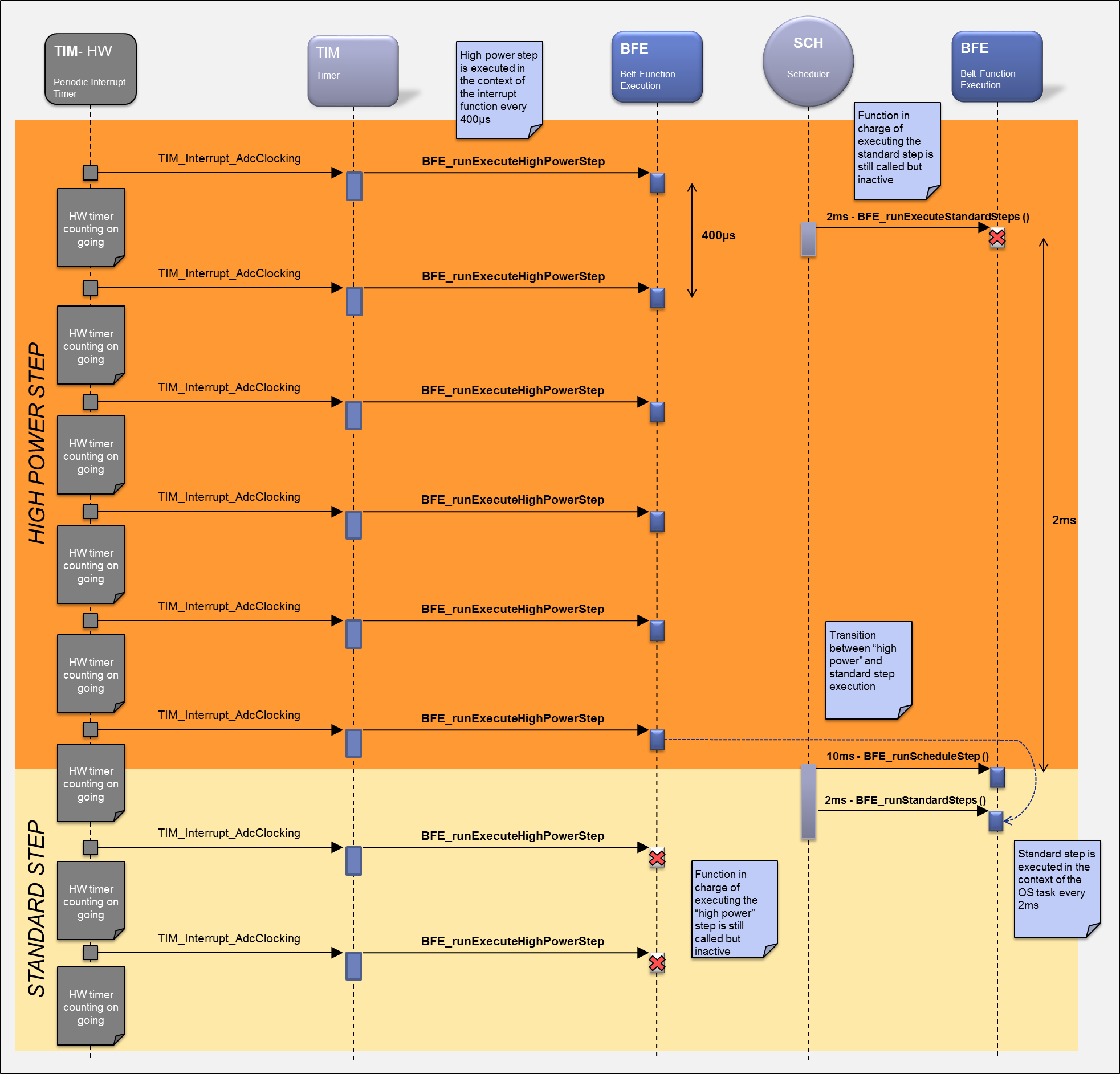
For these reasons, the “high power” step will be executed by the BFE\_runExecuteHighPowerStep function in the context of the TIM\_Interrupt\_AdcClocking interrupt function.



**Figure 9: Bfe - Data flow description to execute the high power step**

Note:

As shown by the figure below BFE\_runExecuteStandardSteps and BFE\_runExecuteHighPowerStep functions are always called in parallel. Nevertheless only one of these 2 functions are called, depending on the step parameters.

****

**Figure 10: Bfe - Dynamic description to execute a cycle with booster function**

## To execute the emergency motor braking sequence (eCPL function)

### Principle

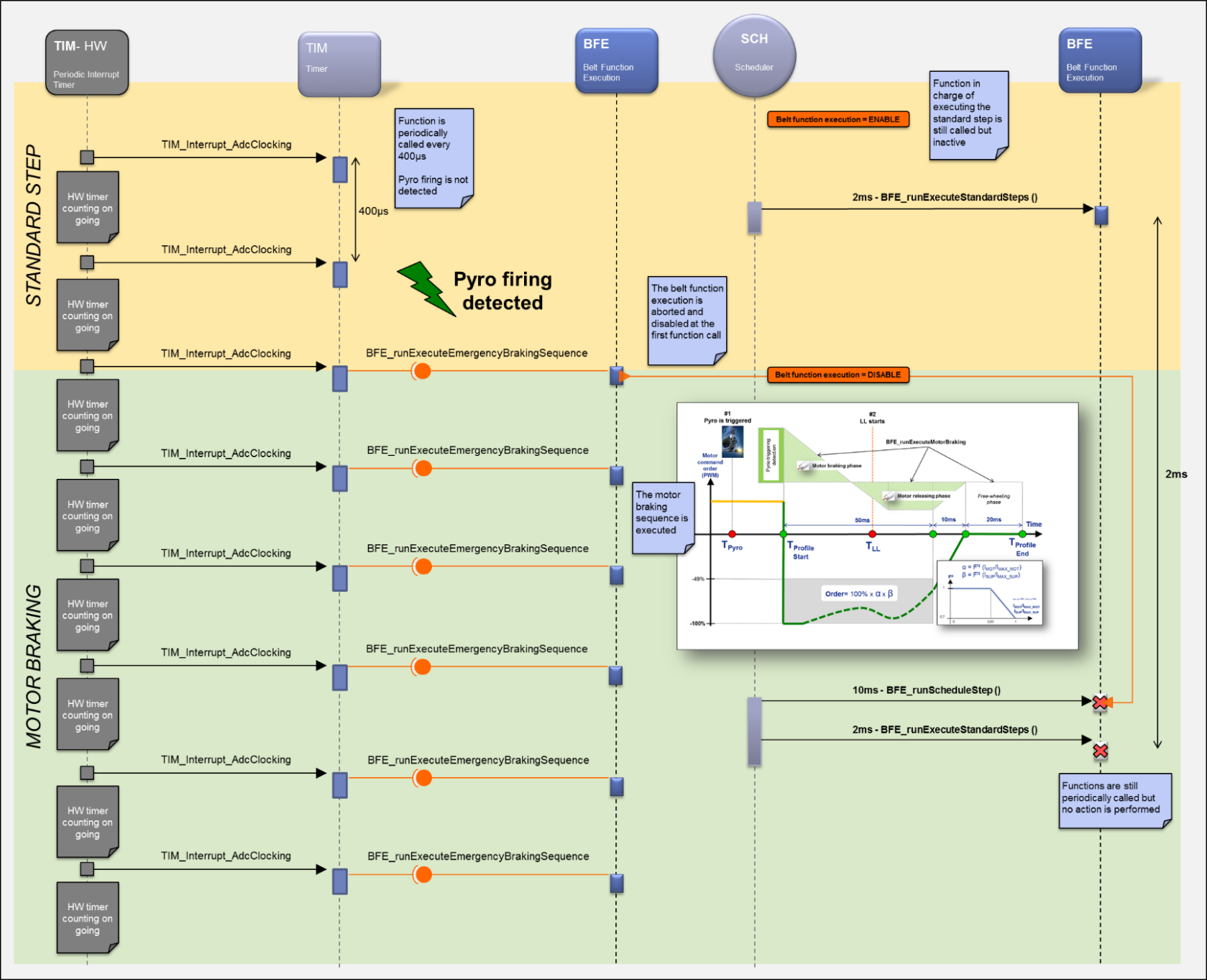
The execution of the emergency motor braking sequence is based on 1 function:

* BFE\_runExecuteEmergencyBrakingSequence:
  + To disable the belt function activation until the next ignition cycle
  + To brake the motor (if motor activation is on-going)
  + Execution period: 400µs

This function will be executed as soon as the pyro device firing is detected (refer to the “**Error! Reference source not found.**” chapter to get an overiview).

### To disable the belt function activation

As mentioned before the first task of the BFE\_runExecuteEmergencyBrakingSequence function is to disable the belt function activation until the next ignition cycle. This will be done based on a BFE internal data.

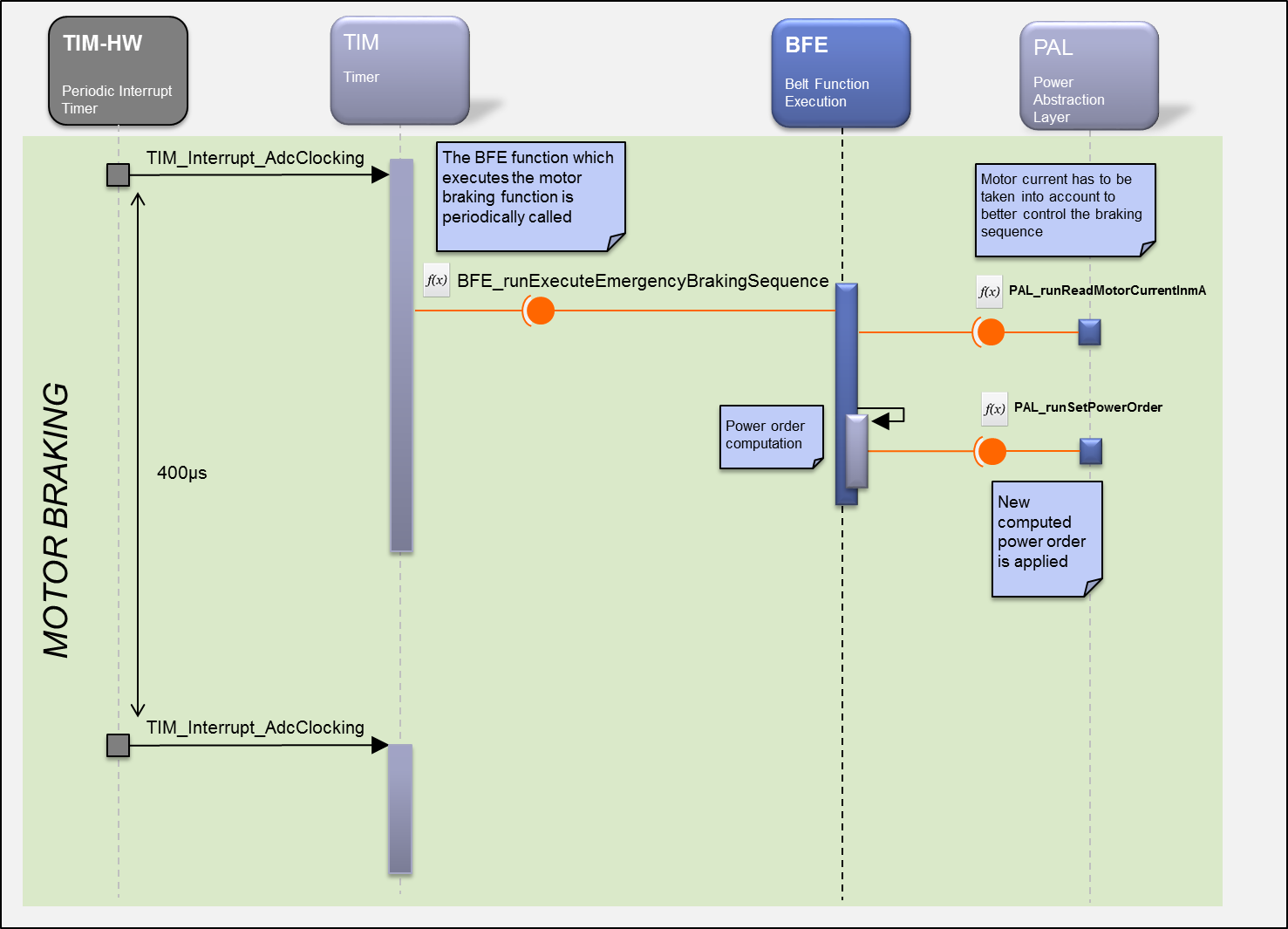


**Figure 11: Bfe – To disable the belt functions**

### To brake the motor rotation

If the pyro firing is detected whereas the motor is activated then the PP ECU will execute a motor braking sequence to stop the motor as fast as possible.

Therefore, the emergency braking sequence will be performed in the context of the TIM\_Interrupt\_AdcClocking interrupt function, to guarantee the best response time for the motor control (like for the “high power” step execution).



**Figure 12: Bfe - To execute the emergency motor braking sequence**

## To update the power supply under critical section

As described in the previous sections the HW power stage is controlled by different BFE processes:

* The standard step execution
* The high power step execution
* The motor braking sequence

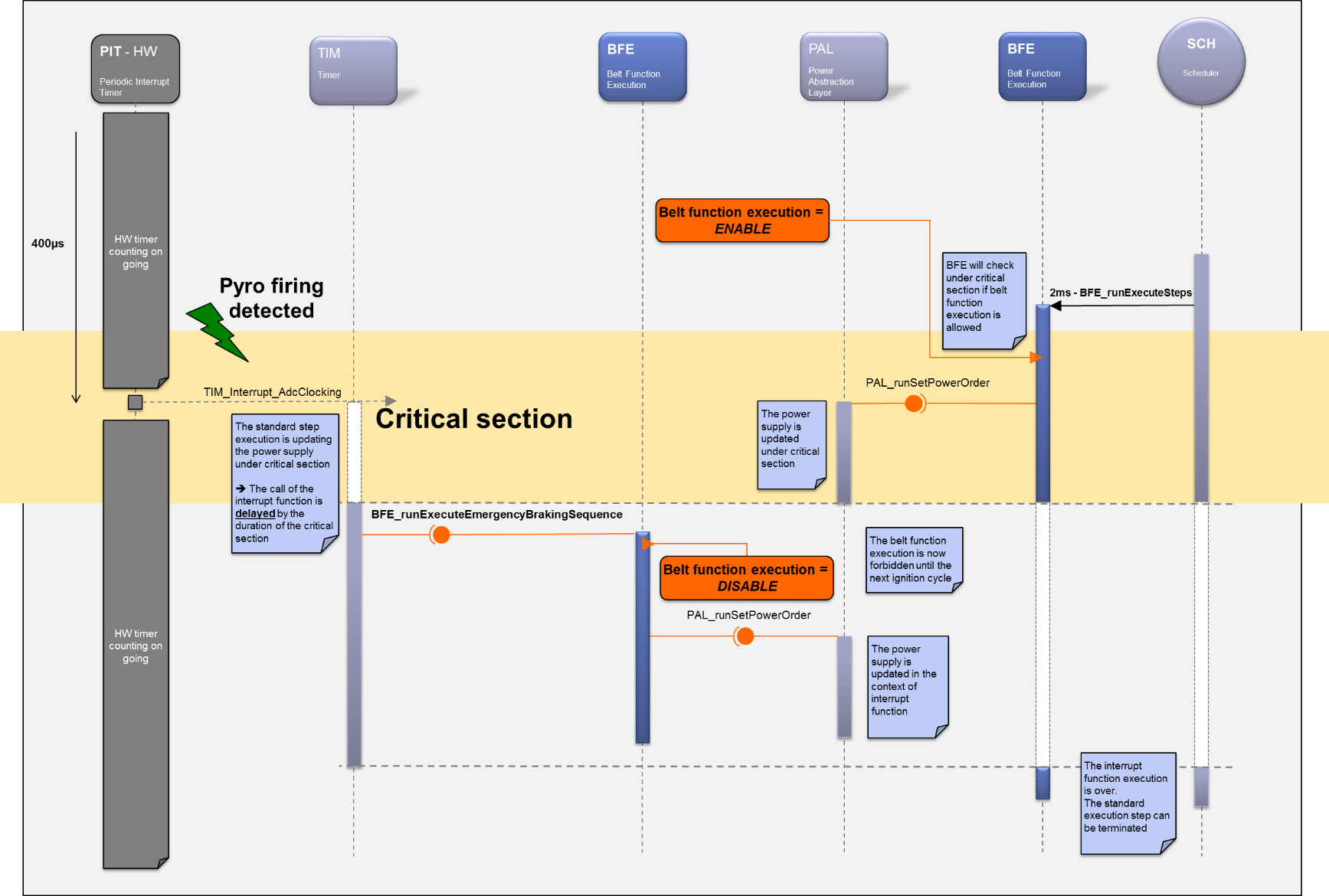
Since these processes are executed in different real time contexts (scheduler and interrupt function), a critical section will be defined to prevent from concurrent access to the HW power stage.

Moreover, this critical section will also be used to prevent from executing a belt function (at BFE level) once the pyro firing is detected.

Below are examples to illustrate some specific real time cases for which the use of this critical section is a must.

The first case occurs when the standard step execution updates the power supply whereas the motor braking sequence needs to start at the same time (due to pyro firing detection).

In this particular case, the critical section will allow postponing the call of the interrupt function after the update of the power supply done by the standard step execution (see the figure below).

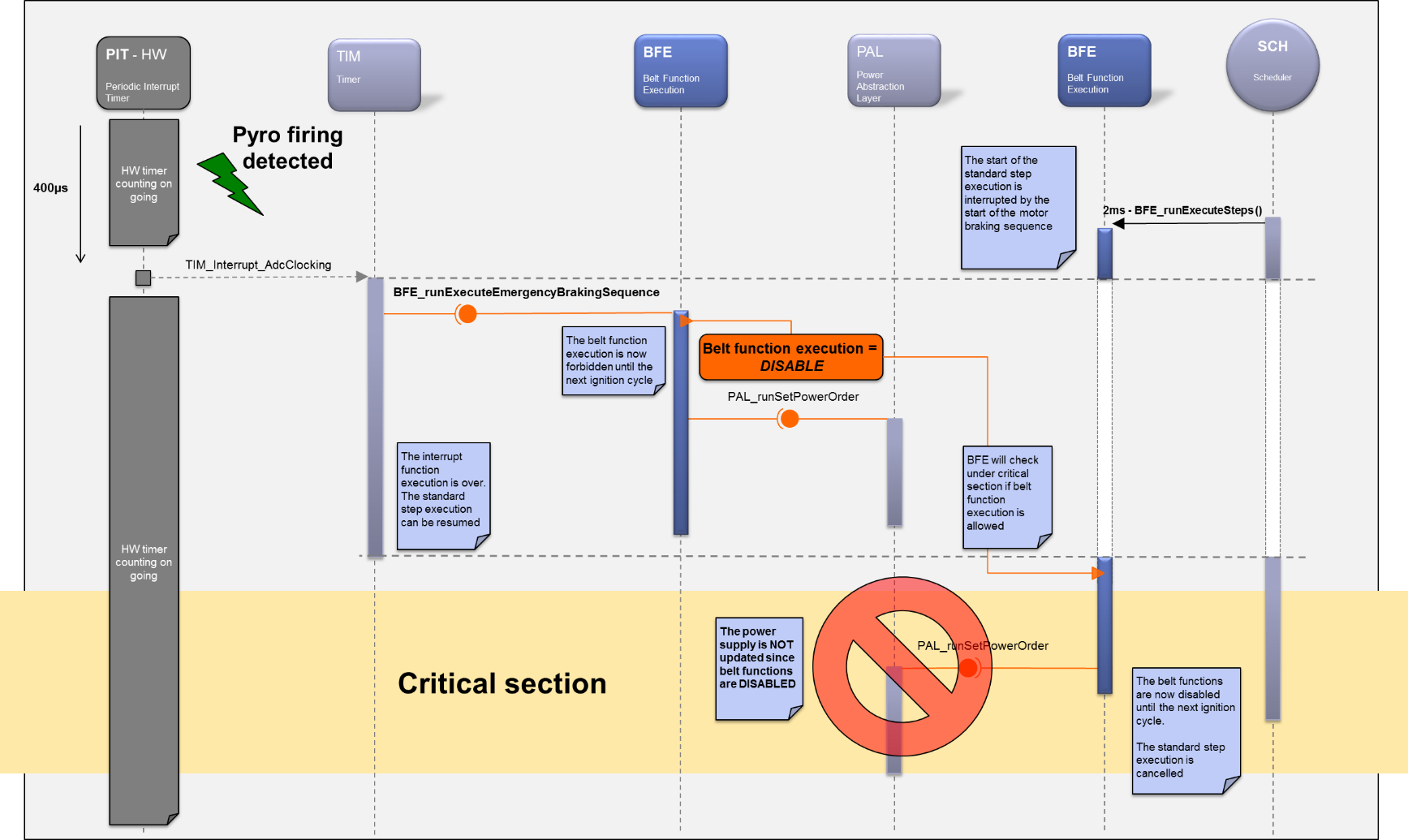
****

**Figure 13: Bfe - To update the motor power order under critical section – Case #1**

The second case occurs when the standard step execution is interrupted by the start of the motor braking sequence before updating the power supply.

In this particular case, the motor braking sequence will deactivate the belt function whereas the update of the power supply is pending at standard step execution level.

Back to the interrupted standard step execution, the update of the power supply (pending) will be cancelled since the belt functions are now disabled.



**Figure 14: Bfe - To update the motor power order under critical section – Case #2**

# Runnables

## BFE\_runScheduleStep

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **BFE\_runScheduleStep** (void) | | | |
| **Object** | | | |
| This function shall manage the start, the execution and the stop of the belt function:   * Cycle parameters extraction * Steps scheduling * Steps parameters extraction/computation * … | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Periodic  Non Reentrant | | | |
| **Requirements** | | | |
| ARCH\_SW\_BFE\_0300 | | | |
| **Covered requirements** | | | |
| ALV\_EXT\_TF\_G\_789, ALV\_EXT\_TF\_G\_790, ALV\_EXT\_TF\_G\_16, ALV\_EXT\_TF\_G\_17, ALV\_EXT\_TF\_G\_18, ALV\_EXT\_TF\_G\_19, ALV\_EXT\_TF\_G\_21, ALV\_EXT\_TF\_G\_22, ALV\_EXT\_TF\_G\_25, ALV\_EXT\_TF\_G\_26, ALV\_EXT\_TF\_G\_27, ALV\_EXT\_TF\_G\_28, ALV\_EXT\_TF\_G\_29, ALV\_EXT\_TF\_G\_66, ALV\_EXT\_TF\_G\_67, ALV\_EXT\_TF\_G\_68, ALV\_EXT\_TF\_G\_69, ALV\_EXT\_TF\_G\_70, ALV\_EXT\_TF\_G\_71, ALV\_EXT\_TF\_G\_72, ALV\_EXT\_TF\_G\_73, ALV\_EXT\_TF\_G\_78, ALV\_EXT\_TF\_G\_79, ALV\_EXT\_TF\_G\_80, ALV\_EXT\_TF\_G\_82, ALV\_EXT\_TF\_G\_83, ALV\_EXT\_TF\_G\_84, ALV\_EXT\_TF\_G\_85, ALV\_EXT\_TF\_G\_86, ALV\_EXT\_TF\_G\_87, ALV\_EXT\_TF\_G\_91, ALV\_EXT\_TF\_G\_92, ALV\_EXT\_TF\_G\_93, ALV\_EXT\_TF\_G\_94, ALV\_EXT\_TF\_G\_95, ALV\_EXT\_TF\_G\_96, ALV\_EXT\_TF\_G\_97, ALV\_EXT\_TF\_G\_98, ALV\_EXT\_TF\_G\_99, ALV\_EXT\_TF\_G\_100, ALV\_EXT\_TF\_G\_101, ALV\_EXT\_TF\_G\_102, ALV\_EXT\_TF\_G\_103, ALV\_EXT\_TF\_G\_104, ALV\_EXT\_TF\_G\_105, ALV\_EXT\_TF\_G\_106, ALV\_EXT\_TF\_G\_313, ALV\_EXT\_TF\_G\_420, ALV\_EXT\_TF\_G\_421, ALV\_EXT\_TF\_G\_422, ALV\_EXT\_TF\_G\_426, ALV\_EXT\_TF\_G\_427, ALV\_EXT\_TF\_G\_430, ALV\_EXT\_TF\_G\_649, ALV\_EXT\_TF\_G\_650, ALV\_EXT\_TF\_G\_652, ALV\_EXT\_TF\_G\_653, ALV\_EXT\_TF\_G\_654, ALV\_EXT\_TF\_G\_655, ALV\_EXT\_TF\_G\_656, ALV\_EXT\_TF\_G\_657, ALV\_EXT\_TF\_G\_658, ALV\_EXT\_TF\_G\_659, ALV\_EXT\_TF\_G\_661, ALV\_EXT\_TF\_G\_663, ALV\_EXT\_TF\_G\_664, ALV\_EXT\_TF\_G\_665, ALV\_EXT\_TF\_G\_668, ALV\_EXT\_TF\_G\_669, ALV\_EXT\_TF\_G\_672, ALV\_EXT\_TF\_G\_673, ALV\_EXT\_TF\_G\_674, ALV\_EXT\_TF\_G\_675, ALV\_EXT\_TF\_G\_676, ALV\_EXT\_TF\_G\_677, ALV\_EXT\_TF\_G\_683, ALV\_EXT\_TF\_G\_684, ALV\_EXT\_TF\_G\_685, ALV\_EXT\_TF\_G\_686, ALV\_EXT\_TF\_G\_689, ALV\_EXT\_TF\_G\_692, ALV\_EXT\_TF\_G\_693, ALV\_EXT\_TF\_G\_88, ALV\_EXT\_TF\_R\_113, ALV\_EXT\_TF\_R\_112, ALV\_EXT\_TF\_R\_116, ALV\_EXT\_TF\_R\_117 , ALV\_EXT\_TF\_R\_2344 | | | |

### Data flow

#### Input Data

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_0050 | The BFE\_runScheduleStep function shall read every 10ms the **selected belt function** provided by the BFS component.  This is the main input of the BFE component containing the selected belt function which has to be executed. |  | ALV\_EXT\_TF\_G\_32, ALV\_EXT\_TF\_G\_311,  ALV\_EXT\_TF\_G\_886,  TR6\_EXT\_TF\_G\_913 |
| ARCH\_SW\_BFE\_0051 | The AUTOSAR read access explicit mode shall be used between these 2 components to read the **selected belt function**.  Justification:  This explicit mode is necessary since the BFE and BFS main functions are mapped into the same OS task.  Otherwise, the belt function execution start will be delayed by 10ms. |  |  |
| ARCH\_SW\_BFE\_0052 | The **executed step number** data shall be initialized to “NO\_STEP” by default. |  | ALV\_EXT\_TF\_G\_54, ALV\_EXT\_TF\_G\_55, ALV\_EXT\_TF\_G\_56 |
| ARCH\_SW\_BFE\_0053 | The BFE\_runScheduleSteps function shall use the **NVP\_au8BeltProfilesDefinitions** macro to read the cycle parameters from the NVP component. |  | ALV\_EXT\_TF\_G\_395, ALV\_EXT\_TF\_G\_7, ALV\_EXT\_TF\_G\_36, ALV\_EXT\_TF\_G\_308,  ALV\_EXT\_TF\_G\_9,  ALV\_EXT\_TF\_G\_35 |
| ARCH\_SW\_BFE\_0054 | The BFE\_runScheduleSteps function shall use the **NVP\_au8StepsDefinitions** macro to read the steps parameters from the NVP component. |  | ALV\_EXT\_TF\_G\_395, ALV\_EXT\_TF\_G\_7, ALV\_EXT\_TF\_G\_8, ALV\_EXT\_TF\_G\_36, ALV\_EXT\_TF\_G\_305, ALV\_EXT\_TF\_G\_306,  ALV\_EXT\_TF\_G\_38,  ALV\_EXT\_TF\_G\_75 |
| ARCH\_SW\_BFE\_0055 | The BFE component shall use the **NVP\_au16CRCCyleParameter** macro to read the checksum related to the cycle and step parameters from the NVP component. |  | ALV\_EXT\_TF\_G\_395, ALV\_EXT\_TF\_G\_7, ALV\_EXT\_TF\_G\_36 |
| ARCH\_SW\_BFE\_0056 | The **executed belt function** (aka cycle number) data shall be an input. |  | ALV\_EXT\_TF\_G\_401,  ALV\_EXT\_TF\_R\_102 |
| ARCH\_SW\_BFE\_0057 | The **Temperature compensation table** parametershall be an input. |  | ALV\_EXT\_TF\_G\_411, ALV\_EXT\_TF\_G\_412, ALV\_EXT\_TF\_G\_413,  ALV\_EXT\_TF\_G\_415,  ALV\_EXT\_TF\_G\_416 |

#### Output Data

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_0100 | The BFE\_runScheduleStep function shall write every 10ms the **executed belt function** (aka cycle number).  This is one of the main outputs of the BFE component to inform the rest of the application of the current executed belt function. |  | ALV\_EXT\_TF\_G\_43, ALV\_EXT\_TF\_G\_45 |
| ARCH\_SW\_BFE\_0101 | The **executed belt function** data (aka cycle number)shall be initialized to “NO\_CYCLE” by default. |  | ALV\_EXT\_TF\_G\_46, ALV\_EXT\_TF\_G\_47, ALV\_EXT\_TF\_G\_48,  ALV\_EXT\_TF\_R\_105,  ALV\_EXT\_TF\_R\_106,  ALV\_EXT\_TF\_R\_109,  ALV\_EXT\_TF\_R\_110 |
| ARCH\_SW\_BFE\_0102 | The BFE\_runScheduleStep function shall write every 10ms the **executed step number** data. |  | ALV\_EXT\_TF\_G\_50, ALV\_EXT\_TF\_G\_52, ALV\_EXT\_TF\_G\_53 |
| ARCH\_SW\_BFE\_0103 | **Step duration** shall be provided by this function |  | ALV\_EXT\_TF\_G\_317, ALV\_EXT\_TF\_G\_318 |
| ARCH\_SW\_BFE\_0104 | **Step order type** shall be provided by this function |  | ALV\_EXT\_TF\_G\_321 |
| ARCH\_SW\_BFE\_0105 | **Step order value** shall be provided by this function |  | ALV\_EXT\_TF\_G\_324, ALV\_EXT\_TF\_G\_326, ALV\_EXT\_TF\_G\_327, ALV\_EXT\_TF\_G\_329, ALV\_EXT\_TF\_G\_331, ALV\_EXT\_TF\_G\_333, ALV\_EXT\_TF\_G\_335, ALV\_EXT\_TF\_G\_337,  ALV\_EXT\_TF\_G\_887,  ALV\_EXT\_TF\_G\_883 |
| ARCH\_SW\_BFE\_0106 | **Trig Off option** shall be provided by this function |  | ALV\_EXT\_TF\_G\_340 ,  ALV\_EXT\_TF\_G\_688 |
| ARCH\_SW\_BFE\_0107 | **Ramp Step option** shall be provided by this function |  | ALV\_EXT\_TF\_G\_343, ALV\_EXT\_TF\_G\_344 |
| ARCH\_SW\_BFE\_0108 | **Current regulation type** shall be provided by this function |  | ALV\_EXT\_TF\_G\_347, ALV\_EXT\_TF\_G\_349 |
| ARCH\_SW\_BFE\_0109 | **Belt movement detection option** shall be provided by this function |  | ALV\_EXT\_TF\_G\_351 |
| ARCH\_SW\_BFE\_0110 | **Blocked Belt option** shall be provided by this function |  | ALV\_EXT\_TF\_G\_356 |
| ARCH\_SW\_BFE\_0111 | **Motor blocked detection option** shall be provided by this function |  | ALV\_EXT\_TF\_G\_361 |
| ARCH\_SW\_BFE\_0112 | **Release controlled step option** shall be provided by this function |  | ALV\_EXT\_TF\_G\_365 |
| ARCH\_SW\_BFE\_0113 | **Weigthed step order option** shall be provided by this function |  | ALV\_EXT\_TF\_G\_369, ALV\_EXT\_TF\_G\_370, ALV\_EXT\_TF\_G\_371, ALV\_EXT\_TF\_G\_372, ALV\_EXT\_TF\_G\_373, ALV\_EXT\_TF\_G\_374 |
| ARCH\_SW\_BFE\_0114 | **Adaptative current limitation option** shall be provided by this function |  | ALV\_EXT\_TF\_G\_377, ALV\_EXT\_TF\_G\_379, ALV\_EXT\_TF\_G\_380, ALV\_EXT\_TF\_G\_381 |
| ARCH\_SW\_BFE\_0115 | **Belt pay out abortion option** shall be provided by this function |  | ALV\_EXT\_TF\_G\_384, ALV\_EXT\_TF\_G\_386 |
| ARCH\_SW\_BFE\_0116 | **Step validty flag** shall be provided by this function |  | ALV\_EXT\_TF\_G\_388 |
| ARCH\_SW\_BFE\_0117 | **Power degradation factor** shall be provided by this function | Unsigned type  1LSB ~100/128% | ALV\_EXT\_TF\_G\_404, ALV\_EXT\_TF\_G\_406, ALV\_EXT\_TF\_G\_407 |
| ARCH\_SW\_BFE\_0118 | **Temperature compensation type** shall be provided by this function |  | ALV\_EXT\_TF\_G\_410,  ALV\_EXT\_TF\_G\_483 |
| ARCH\_SW\_BFE\_0119 | **Velocity control step order type** shall be provided by this function |  | TR6\_EXT\_TF\_G\_967,  TR6\_EXT\_TF\_G\_952 ; TR6\_EXT\_TF\_G\_953 ;  TR6\_EXT\_TF\_G\_954 |
| ARCH\_SW\_BFE\_0120 | **Velocity control step order value** shall be provided by this function |  | TR6\_EXT\_TF\_G\_966 |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_00300 | The BFE\_runScheduleStep function shall be periodically called every 10ms.  This function will be responsible for:   * Starting a new cycle if required * Scheduling steps execution |  | ALV\_EXT\_TF\_G\_11 |
| ARCH\_SW\_BFE\_0083 | The BFE\_runScheduleSteps function shall call the PMP\_runGetBatteryVoltage function.  It will allow computing the power degradation ratio. |  | ALV\_EXT\_TF\_G\_398 |
| ARCH\_SW\_BFE\_0453 | [PMP\_runGetFilteredTemperature](#_Hlk412205769) shall be called to get the temperature measured at the HW sensor.  This information is used by the temperature compensation feature. | HW sensor is located in the Hearst MCU for the mainstream variant of the PP4G product | ALV\_EXT\_TF\_G\_481 |
| ARCH\_SW\_BFE\_0455 | [BMM\_runGetSpeed\_mm\_s](#_Hlk435681986) shall be called to get the belt speed. |  | ALV\_EXT\_TF\_G\_465 |
| ARCH\_SW\_BFE\_0456 | [BMM\_runGetPositionFromT0\_mm](#_Hlk457182758) shall be called to get the position of the belt. |  | ALV\_EXT\_TF\_G\_470 |

## BFE\_runExecuteStandardSteps

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **BFE\_runExecuteStandardSteps** (void) | | | |
| **Object** | | | |
| This function shall execute the standard step of cycle (without booster function). | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Periodic: 2 ms  Non Reentrant | | | |
| **Requirements** | | | |
| ARCH\_SW\_BFE\_0370 | | | |
| **Covered requirements** | | | |
| ALV\_EXT\_TF\_G\_789, ALV\_EXT\_TF\_G\_790, ALV\_EXT\_TF\_G\_522, ALV\_EXT\_TF\_G\_523, ALV\_EXT\_TF\_G\_524, ALV\_EXT\_TF\_G\_525, ALV\_EXT\_TF\_G\_526, ALV\_EXT\_TF\_G\_529, ALV\_EXT\_TF\_G\_530, ALV\_EXT\_TF\_G\_532, ALV\_EXT\_TF\_G\_533, ALV\_EXT\_TF\_G\_534, ALV\_EXT\_TF\_G\_535, ALV\_EXT\_TF\_G\_540, ALV\_EXT\_TF\_G\_541, ALV\_EXT\_TF\_G\_542, ALV\_EXT\_TF\_G\_543, ALV\_EXT\_TF\_G\_544, ALV\_EXT\_TF\_G\_552, ALV\_EXT\_TF\_G\_553, ALV\_EXT\_TF\_G\_554, ALV\_EXT\_TF\_G\_556, ALV\_EXT\_TF\_G\_557, ALV\_EXT\_TF\_G\_564, ALV\_EXT\_TF\_G\_565, ALV\_EXT\_TF\_G\_566, ALV\_EXT\_TF\_G\_567, ALV\_EXT\_TF\_G\_569, ALV\_EXT\_TF\_G\_570, ALV\_EXT\_TF\_G\_571, ALV\_EXT\_TF\_G\_573, ALV\_EXT\_TF\_G\_574, ALV\_EXT\_TF\_G\_583, ALV\_EXT\_TF\_G\_584, ALV\_EXT\_TF\_G\_604, ALV\_EXT\_TF\_G\_605, ALV\_EXT\_TF\_G\_606, ALV\_EXT\_TF\_G\_611, ALV\_EXT\_TF\_G\_612, ALV\_EXT\_TF\_G\_545, ALV\_EXT\_TF\_G\_546, ALV\_EXT\_TF\_G\_547, ALV\_EXT\_TF\_G\_548, ALV\_EXT\_TF\_G\_549, ALV\_EXT\_TF\_G\_550, ALV\_EXT\_TF\_G\_652, ALV\_EXT\_TF\_G\_713, ALV\_EXT\_TF\_G\_715, ALV\_EXT\_TF\_G\_716, ALV\_EXT\_TF\_G\_717, ALV\_EXT\_TF\_G\_718, ALV\_EXT\_TF\_G\_608, ALV\_EXT\_TF\_G\_610, | | | |

### Data flow

#### Input Data

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_0371 | **Step duration** shall be an input |  | ALV\_EXT\_TF\_G\_435 |
| ARCH\_SW\_BFE\_0372 | **Step order type** shall be an input. |  | ALV\_EXT\_TF\_G\_437 |
| ARCH\_SW\_BFE\_0373 | **Step order value** shall be an input. |  | ALV\_EXT\_TF\_G\_439 |
| ARCH\_SW\_BFE\_0374 | **Trig Off option** shall be an input. |  | ALV\_EXT\_TF\_G\_447 |
| ARCH\_SW\_BFE\_0375 | **Ramp Step option** shall be an input. |  | ALV\_EXT\_TF\_G\_455 |
| ARCH\_SW\_BFE\_0376 | **Current regulation type** shall be an input. |  | ALV\_EXT\_TF\_G\_457 |
| ARCH\_SW\_BFE\_0377 | **Belt movement detection option** shall be an input. |  | ALV\_EXT\_TF\_G\_443 |
| ARCH\_SW\_BFE\_0378 | **Blocked Belt option** shall be an input. |  | ALV\_EXT\_TF\_G\_449 |
| ARCH\_SW\_BFE\_0379 | **Motor blocked detection option** shall be an input. |  | ALV\_EXT\_TF\_G\_451 |
| ARCH\_SW\_BFE\_0380 | **Release controlled step option** shall be an input. |  | ALV\_EXT\_TF\_G\_453 |
| ARCH\_SW\_BFE\_0381 | **Weigthed step order option** shall be an input. |  | ALV\_EXT\_TF\_G\_445 |
| ARCH\_SW\_BFE\_0382 | **Adaptative current limitation option** shall be an input. |  | ALV\_EXT\_TF\_G\_441 |
| ARCH\_SW\_BFE\_0383 | **Belt pay out abortion option** shall be an input. |  |  |
| ARCH\_SW\_BFE\_0384 | **Step validty flag** shall be an input. |  | ALV\_EXT\_TF\_G\_459 |
| ARCH\_SW\_BFE\_0385 | **Power degradation factor** shall be an input. | Unsigned type  1LSB ~100/128% |  |
| ARCH\_SW\_BFE\_0386 | **Temperature compensation type** shall be an input. |  |  |
| ARCH\_SW\_BFE\_0387 | The **belt function activation status** (internal runnable variable of the BFE) shall be read.  It the belt functions are deactivated then the HW power stage control will not be permitted. |  | ALV\_EXT\_TF\_G\_709 |
| ARCH\_SW\_BFE\_0388 | The **trig OFF status** shall be an input. |  | ALV\_EXT\_TF\_G\_487 |
| ARCH\_SW\_BFE\_0389 | The **PWM offset parameter** shall be an input. |  | ALV\_EXT\_TF\_G\_571 |
| ARCH\_SW\_BFE\_0390 | The **max current value parameter** shall be an input. |  | ALV\_EXT\_TF\_G\_574 |
| ARCH\_SW\_BFE\_0391 | The **PID factors parameter** shall be an input. |  | ALV\_EXT\_TF\_G\_585,  TR6\_EXT\_TF\_G\_915 |
| ARCH\_SW\_BFE\_0392 | The **Motor current** shall be an input. |  | ALV\_EXT\_TF\_G\_475 |
| ARCH\_SW\_BFE\_0393 | **Velocity control step order type** shall be an input |  | TR6\_EXT\_TF\_G\_968 ; TR6\_EXT\_TF\_G\_972 ;  TR6\_EXT\_TF\_G\_976 ;  TR6\_EXT\_TF\_G\_979 |
| ARCH\_SW\_BFE\_0394 | **Velocity control step order value** shall be an input |  | TR6\_EXT\_TF\_G\_927 ; ALV\_EXT\_TF\_G\_963 |

#### Output Data

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_0400 | The **step ending flag** shall be provided by this function |  | ALV\_EXT\_TF\_G\_509, ALV\_EXT\_TF\_G\_511 |
| ARCH\_SW\_BFE\_0401 | The **velocity control loop algorithm** shall be implemeneted by this function. |  | ALV\_EXT\_TF\_G\_967;  ALV\_EXT\_TF\_G\_967;  TR6\_EXT\_TF\_G\_982;  TR6\_EXT\_TF\_G\_981 |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_0450 | [PAL\_runSetPowerOrder](#_Hlk457181360) shall be called to apply the new computed motor power order. |  | ALV\_EXT\_TF\_G\_774, ALV\_EXT\_TF\_G\_796, ALV\_EXT\_TF\_G\_493,  ALV\_EXT\_TF\_G\_884 |
| ARCH\_SW\_BFE\_0451 | [PAL\_runSetPowerOrder](#_Hlk457181360) call shall be performed under critical section. |  | ALV\_EXT\_TF\_G\_774, ALV\_EXT\_TF\_G\_796,  ALV\_EXT\_TF\_G\_884 |
| ARCH\_SW\_BFE\_0452 | [PAL\_runReadMotorCurrentInmA](#_Hlk424738645) shall be called to get the motor current value in mA from PAL.  This information will be used for the steps with motor current regulation. |  | ALV\_EXT\_TF\_G\_475 |
| ARCH\_SW\_BFE\_0454 | [PMP\_runGetBatteryVoltage](#_Hlk411940329) shall be called. |  | ALV\_EXT\_TF\_G\_461 |

## BFE\_runExecuteHighPowerStep

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **BFE\_runExecuteHighPowerStep** (void) | | | |
| **Object** | | | |
| This function shall execute the “high power” step of cycle. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Periodic  Non Reentrant | | | |
| **Requirements** | | | |
| ARCH\_SW\_BFE\_0500 | | | |
| **Covered requirements** | | | |
| ALV\_EXT\_TF\_G\_789, ALV\_EXT\_TF\_G\_790, ALV\_EXT\_TF\_G\_616, ALV\_EXT\_TF\_G\_621, ALV\_EXT\_TF\_G\_623, ALV\_EXT\_TF\_G\_626, ALV\_EXT\_TF\_G\_629, ALV\_EXT\_TF\_G\_630, ALV\_EXT\_TF\_G\_634, ALV\_EXT\_TF\_G\_635, ALV\_EXT\_TF\_G\_636, ALV\_EXT\_TF\_G\_637, ALV\_EXT\_TF\_G\_638, ALV\_EXT\_TF\_G\_639, ALV\_EXT\_TF\_G\_641, ALV\_EXT\_TF\_G\_642, ALV\_EXT\_TF\_G\_643, ALV\_EXT\_TF\_G\_644 | | | |

### Data flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_0501 | The **belt function activation status** (internal runnable variable of the BFE) shall be read.  It the belt functions are deactivated then the HW power stage control will not be possible. |  | ALV\_EXT\_TF\_G\_709 |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_0202 | [PAL\_runReadMotorCurrentInmA](#_Hlk424738645) shall be called to get the motor current value in mA from PAL.  This information will be used for the steps with motor current regulation. |  | ALV\_EXT\_TF\_G\_11 |
| ARCH\_SW\_BFE\_0203 | [PMP\_runGetBatteryVoltage](#_Hlk411940329) shall be called. |  | ALV\_EXT\_TF\_G\_509 |
| ARCH\_SW\_BFE\_0204 | [PAL\_runSetPowerOrder](#_Hlk457181360) shall be called to apply the new computed motor power order. |  | ALV\_EXT\_TF\_G\_501 |
| ARCH\_SW\_BFE\_0205 | [PAL\_runSetPowerOrder](#_Hlk457181360) call shall be performed under critical section.  Actually, the critical section is implicit in the SW since:   * This function is called in the context of the interrupt function   Nested interrupts are not allowed |  | ALV\_EXT\_TF\_G\_774, ALV\_EXT\_TF\_G\_796,  ALV\_EXT\_TF\_G\_880, |
| ARCH\_SW\_BFE\_0206 | The integrity of the BFE data shall be ensured between the execution of the high power and standard steps.  This will be particular the case when entering or leaving the high power step execution. |  |  |
| ARCH\_SW\_BFE\_0207 | [PMP\_runGetMotorVp](#_Hlk424200200) shall be called. |  | ALV\_EXT\_TF\_G\_629 |

### Miscelaneous

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_220 | This function shall be implemented under preprocessor option. |  | ALV\_EXT\_TF\_G\_617 |

## BFE\_runExecuteEmergencyBrakingSequence

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **BFE\_runExecuteEmergencyBrakingSequence** (void) | | | |
| **Object** | | | |
| This function shall manage the emergency braking sequence of the motor. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Periodic  Non Reentrant | | | |
| **Requirements** | | | |
| ARCH\_SW\_BFE\_0600 | | | |
| **Covered requirements** | | | |
| ALV\_EXT\_TF\_G\_789, ALV\_EXT\_TF\_G\_790, ALV\_EXT\_TF\_G\_716, ALV\_EXT\_TF\_G\_723, ALV\_EXT\_TF\_G\_724, ALV\_EXT\_TF\_G\_743, ALV\_EXT\_TF\_G\_761, ALV\_EXT\_TF\_G\_762, ALV\_EXT\_TF\_G\_763; ALV\_EXT\_TF\_G\_908, ALV\_EXT\_TF\_G\_726, ALV\_EXT\_TF\_G\_744 | | | |

### Data flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_0601 | NVP\_au16AlphaMotorCurrent shall be used for the motor power order computation |  | ALV\_EXT\_TF\_G\_725;  ALV\_EXT\_TF\_G\_726 |
| ARCH\_SW\_BFE\_0602 | NVP\_au16BetaSupplyCurrent shall be used for the motor power order computation |  | ALV\_EXT\_TF\_G\_725;  ALV\_EXT\_TF\_G\_744 |
| ARCH\_SW\_BFE\_0603 | The pyro ignition status shall be an input |  | ALV\_EXT\_TF\_G\_908 |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_0260 | The belt function execution status internal data shall be set ENABLE during the SW initialization. |  | ALV\_EXT\_TF\_G\_711 |
| ARCH\_SW\_BFE\_0261 | The belt function execution statu1407s internal data shall be set to DISABLE at the first function call. |  | ALV\_EXT\_TF\_G\_719, ALV\_EXT\_TF\_G\_715 |
| ARCH\_SW\_BFE\_0280 | [PAL\_runReadMotorCurrentInmA](#_Hlk424738645)shall be called to get the motor current value in mA from PAL.  This information will be used to better control the motor braking sequence. |  | ALV\_EXT\_TF\_G\_700 |
| ARCH\_SW\_BFE\_0281 | [PAL\_runGetPowerOrder](#_Hlk412205833) shall be called to take into account the current applied power order. |  | ALV\_EXT\_TF\_G\_702 |
| ARCH\_SW\_BFE\_0282 | [PAL\_runSetPowerOrder](#_Hlk457181360) shall be called to apply the new computed motor power order. |  | ALV\_EXT\_TF\_G\_705, ALV\_EXT\_TF\_G\_707 |

## BFE\_runResetMotorStageStatus

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **BFE\_runResetMotorStageStatus** (void) | | | |
| **Object** | | | |
| Following the detection of pyro device firing this function shall restore the belt function activation in the same operation cycle. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Periodic  Non Reentrant | | | |
| **Requirements** | | | |
| ARCH\_SW\_BFE\_0620 | | | |
| **Covered requirements** | | | |
|  | | | |

### Data flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
|  |  |  |  |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
|  |  |  |  |

# MCU resources

The following requirements on resource consumption objectives apply to the module/package:

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BFE\_9997 | The ROM size consumed by this component shall not exceed 7.5K bytes. |  | TR6\_EXT\_TF\_B\_2591 |
| ARCH\_SW\_BFE\_9998 | The heap size consumed by this component shall not exceed 500 bytes. |  | TR6\_EXT\_TF\_B\_2592 |